

AMENDMENTS TO THE CLAIMS

Claims 1-28 were filed originally.

Claims 26-28 and 5-9, 10-13, 18-21, and 23-25 were previously canceled as being directed to non-elected subject matter.

Claim 29 was added and claims 2 and 15 were cancelled during prosecution, leaving claims 1, 3-4, 14, 16-17, 22, and 29 pending at the time of this Final Action.

For clarity purposes, these claims are canceled and new claims 30-39 are added. Claims 30-39 are identical to original claims 1-4, 14-17, 22, and 29, respectively, with the exception that claims 22 and 29 are reordered as claims 38 and 39 due to dependency.

Claims 1-29: Canceled.

30. (New) A method for processing an extensible mark up language (XML) document comprising:

parsing the XML document into schema elements and data elements;

converting the schema elements into data type definition (DTD) objects;

validating the data elements using the DTD objects; and

if valid, constructing an in-memory tree representation of the XML document using the data elements.

31. (New) The method of claim 30, wherein the converting comprises:

calling a method in a first application program interface (API); and

as a result of calling the first method, calling one or more methods in a second API to construct the DTD objects.

1 32. (New) The method of claim 30, wherein the converting comprises
2 referencing one or more tables that define the schema elements and associated
3 functions for processing the schema elements.
4

5 33. (New) A computer-readable medium having computer-executable
6 instruction, which when executed by a computer, performs the method of claim
7 30.
8

9 34. (New) An architecture for processing an extensible mark up
10 language (XML) document comprising:
11

12 a parser to parse the XML document into elements including schema
13 elements and data elements;
14

15 a schema node factory, called by the parser, to handle calls to construct a
16 node in an in-memory tree representation of the XML document for the elements;
17 and
18

19 a schema builder, called by the schema node factory, to construct data type
20 definition (DTD) objects used in validating the data elements.
21

22 35. (New) The architecture of claim 34, wherein the schema builder
23 utilizes one or more tables to process the elements, the tables containing
24 information defining a schema for the XML data.
25

 36. (New) A computer implemented with the architecture of claim 34.

 37. (New) A client-server system, comprising:

 a server;

 a client connectable to the server to exchange extensible mark up language
 (XML) documents;

1 at least one of the client and the server implementing the architecture of
2 claim 34.

3 38. (New) The architecture of claim 34, further comprising a
4 validation node factory to evaluate whether the data elements comply with
5 constraints set forth in the DTD objects.

6 39. (New) A system for processing an extensible mark up language
7 (XML) document comprising:

8 means for parsing the XML document into schema elements and data
9 elements;

10 means for converting the schema elements into data type definition (DTD)
11 objects;

12 means for validating the data elements using the DTD objects; and

13 if valid, means for constructing an in-memory tree representation of the
14 XML document using the data elements.